

обладнання; застосування гарячого способу нанесення декоративних плівок на металічні формоутворюючі поверхні обладнання і т.д.

Висновок. Для підвищення естетичних та ергономічних показників спеціального технологічного обладнання має бути посилено авторський нагляд за розробкою конструкторської документації із виготовленням дослідних зразків у відповідності до художньо-конструкторських проектів.

Рішення задачі підвищення ефективності обладнання потребує також постійного вдосконалення організаційної структури, науково-методичних, координаційних та проектних функцій сучасної системи художнього конструювання.

ЛІТЕРАТУРА

1. Бондар О.В. Актуальні аспекти управління інноваційною діяльністю машинобудівних підприємств в період трансформації економіки України // Регіональні перспективи. - 2004. - № 3-4 (22-23). - С. 36-37.
 2. Галушак М.П., Оксентюк А.О., Гевко І.Б. Організація виробництва у прикладах та задачах: Навчальний посібник. – К.: Кондор, 2010. – 214 с.
 3. Кузнецов Ю.М., Луців І.В., Дубиняк С.А. Теорія технічних систем. Навчальний посібник. Тернопіль: ТПІ, 1997. – 310 с.
 4. Мелешина Г.А., Аристов Б.Н. Реинжиниринг как направление автоматизации конструкторско-технологической подготовки производства // Автоматизация и современные технологии. - 2001. - № 3. - С. 34-41.
- Твисс Б. Управление научно-техническими нововведениями: Сокр. пер. с англ. – М.: Экономика, 2005. – 271 с.

УДК 004.451

ВАРИАНТ АППАРАТНОЙ ПОДДЕРЖКИ АЛГОРИТМОВ ПЛАНИРОВАНИЯ И ДИСПЕТЧЕРИЗАЦИИ ЗАДАЧ МНОГОПРОЦЕССОРНОЙ РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

А.И. Мартышкин

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В работе представлен один из возможных вариантов аппаратной поддержки алгоритмов подсистемы планирования и диспетчеризации задач в многопроцессорной реконфигурируемой системе. В ходе проведенного исследования получено, что применение аппаратной поддержки алгоритмов планирования и диспетчеризации дает возможность намного быстрее определять, какая задача из очереди готовых к выполнению должна получить процессорное время следующей, чем программная реализация тех же самых алгоритмов внутри планировщика операционной системы.

Ключевые слова: аппаратная поддержка, реконфигурируемая вычислительная система, процессор, диспетчер задач, планировщик, операционная система.

OPTION OF HARDWARE SUPPORT OF PLANNING ALGORITHMS AND DISPATCHING TASKS OF A MULTIPLE PROCESSOR RECONFIGURABLE COMPUTING SYSTEM

A.I. Martyshkin

Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The paper presents one of the possible options for hardware support of algorithms for the subsystem of scheduling and scheduling tasks in a multiprocessor reconfigurable system. In the course of the study, it was found that the use of hardware support for scheduling and dispatching algorithms makes it possible to much faster determine which task from the ready-to-run queue should receive the next processor time than the software implementation of the same algorithms inside the operating system scheduler.

Keywords: hardware support, reconfigurable computing system, processor, task manager, scheduler, operating system.

Введение. Анализ развития средств вычислительной техники позволяет выявить тенденцию к постепенному переходу от программной реализации ряда выполняемых ЭВМ алгоритмов к аппаратной. Многие зарубежные специалисты считают, что наступает время аппаратной реализации алгоритмов операционных систем [1].

Цель работы. В статье приводится один из вариантов аппаратной поддержки алгоритмов планирования и диспетчеризации задач многопроцессорной реконфигурируемой вычислительной системы, необходимый для повышения быстродействия реконфигурируемой вычислительной системы.

Материал и результаты исследований. Как уже отмечалось ранее [2], в аппаратном диспетчере задач присутствует блок, который отвечает за выбор новой задачи. Данный блок реализует несколько различных дисциплин (вариантов) планирования. Выбор задачи осуществляется за один такт и не зависит от дисциплины планирования. Схема выбора новой задачи состоит из ячеек, каждая из которых связана с одним из регистров очереди готовых задач. Ячейка устроена следующим образом (рисунок 1).

В ячейку поступают данные двух регистров очереди задач. Первый регистр содержит данные на выходе из соседней ячейки, второй регистр подается напрямую из очереди готовых задач. Ячейка производит сравнение этих данных и выдает на выход данные только одного из регистров. Ячейка принимает управляющие сигналы от блока управления, который указывает, над

какими из полей регистров очереди задач проводить операцию сравнения, а также вид операции сравнения (больше или меньше). Также возможно полностью запретить сравнение и отключить вход для данных из соседней ячейки, тогда на выход будет напрямую транслироваться данные регистра очереди задач. Ячейка сравнивает следующие поля регистра очереди задач: PRIORITY, COUNTER, а также возможно сравнение объединенного поля, PRIORITY+COUNTER. Благодаря этому реализуются различные варианты планирования [3]. Как это происходит, будет подробно описано далее. Множество ячеек объединяются в схему, вид которой представлен на рисунке 2. Схема за один такт сравнивает данные всех регистров очереди задач и выдает на выход данные регистра той задачи, которая при заданной дисциплине планирования, является первым кандидатом на обслуживание.

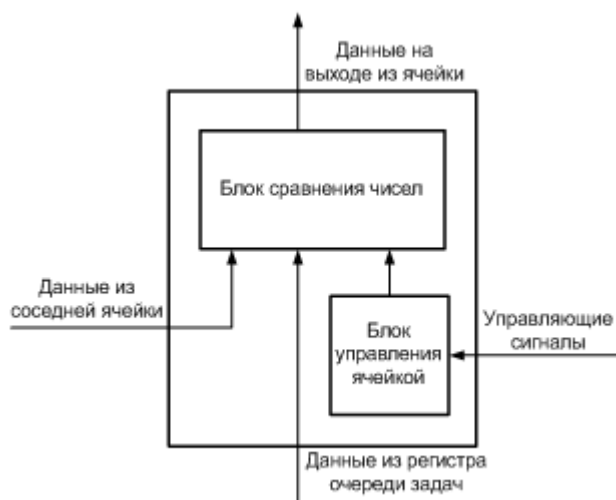


Рисунок 1 – Вид ячейки схемы выбора новой задачи.

Теперь обратимся к различным алгоритмам планирования, используемым в операционных системах, и опишем их реализацию в аппаратном диспетчере задач [4, 5].

Аппаратный диспетчер задач поддерживает следующие варианты планирования: «первым пришел – первым обслужен», «кратчайшая задача – первая», циклическое планирование, приоритетное планирование, приоритетное планирование с классами.

«Первым пришел – первым обслужен».

При таком планировании значения в поле PRIORITY и в поле COUNTER объединяется в одно. Это значение указывает очередность поступления задач в очередь готовых процессов. Схема выбора новой задачи обеспечивает сравнение этого значения у всех регистров очереди задач и выбирает тот регистр, у которого значение в объединенном поле является минимальным. Та-

ким образом, очередь готовых задач работает по принципу памяти FIFO. Первой поступившей задаче присваивается значение 0, следующей 1, и так далее. При этом по умолчанию значения в поле у незанятых регистров очереди равняются максимальному числу, то есть во всех разрядах присутствуют единицы. Также отметим что, в очереди нет регистров с одинаковыми значениями в объединенном поле, кроме свободных регистров.

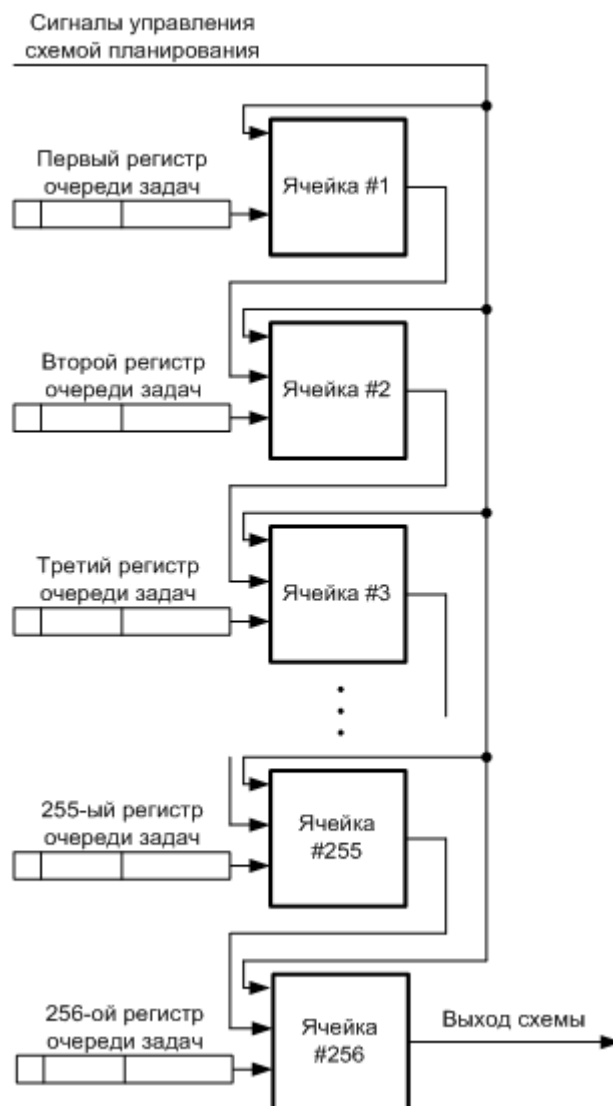


Рисунок 2 – Схема выбора новой задачи.

Отметим что, объединенные поля PRIORITY и COUNTER заполняются аппаратным диспетчером задач.

«Кратчайшая задача – первая».

При планировании такого рода в поле COUNTER записано значение временного отрезка работы процесса. При этом ищется задача, регистр которой содержит минимальное значение в поле COUNTER. Таких задач может быть несколько, тогда они выбираются последовательно. Главным отличием от

реализации предыдущего метода планирования является то, что поле COUNTER формирует не аппаратный диспетчер задач, а операционная система. Диспетчер задач лишь использует эту информацию, для организации такого алгоритма планирования. Определение того, какой из ожидающих процессов является самым коротким, представляется серьезной проблемой, которую невозможно решить аппаратными средствами. Данную задачу решает операционная система. Затем диспетчер задач, основываясь на имеющейся информации, определяет задачу, которая должна идти следующей. У этой задачи значение в поле COUNTER является наименьшим во всей очереди.

Упомянутые алгоритмы планирования используют, как правило, в системах пакетной обработки данных. Рассмотрим алгоритмы планирования, которые применяются в интерактивных системах. Эти алгоритмы могут применяться в качестве планировщика процессоров в системах пакетной обработки.

Циклическое планирование.

Одним из наиболее старых, простых, справедливых и часто используемых является алгоритм циклического планирования. Каждому процессу предоставляется некоторый интервал времени процессора, так называемый квант времени. Если к концу кванта времени процесс все еще работает, он прерывается, а управление передается другому процессу. Планировщику нужно всего лишь поддерживать список процессов в состоянии готовности согласно рисунку 3, а. Когда процесс исчерпал свой лимит времени, он отправляется в конец списка (рисунок 3, б).



Рисунок 3 – Алгоритм циклического планирования: процессы в состоянии готовности (а); процессы в состоянии готовности, когда процесс А исчерпал свой квант времени (б).

Единственным интересным моментом этого алгоритма является длина кванта, значение которого в пределах 20-50 мс зачастую является разумным компромиссом.

Однако, аппаратный диспетчер задач, как уже отмечалось, не контролирует то, сколько задача обслуживается процессором, то есть не выбирает значение кванта и не следит за тем, истекло ли отведенное время. Поэтому

алгоритм циклического планирования реализуется аппаратным диспетчером аналогично алгоритму «первым пришел – первым обслужен». Как и в алгоритме «первым пришел» диспетчер задач формирует очередь готовых задач и задачи отправляются на обслуживание в процессоры в порядке их поступления в очередь. То, сколько по времени обрабатывается задача на процессоре, контролирует программный планировщик.

Ориентируясь по тому, в каком из регистров очереди задач поле COUNTER содержит наименьшее значение, можно организовать аппаратную поддержку алгоритма планирования, называемого справедливым. В этом случае в поле COUNTER фиксируется то, сколько процессорного времени было предоставлено процессу. При этом выбираются те задачи, которые меньше всего обслуживались в процессоре. У них поле COUNTER будет содержать наименьшее значение. При таком планировании система стремится равномерно распределить процессорное время между всеми активными процессами.

Приоритетное планирование.

В циклическом алгоритме планирования есть важное допущение о том, что все процессы равнозначны. Однако есть ситуации, когда обслуживание происходит по приоритетам процессов, которые могут присваиваться статически или динамически.

Приоритет процесса при приоритетном планировании записан в поле PRIORITY регистра задачи. Приоритет процессу назначает операционная система, аппаратный диспетчер задач использует это значение чтобы, выбирать процессы с наивысшим приоритетом. Схема выбора задач отыскивает ту задачу, в регистре которой указано наибольшее значение в поле PRIORITY. Если таких задач несколько их выбор осуществляется случайным образом. Затем выбираются задачи, у которых приоритеты ниже.

Часто бывает удобно сгруппировать процессы в классы по приоритетам и использовать приоритетное планирование среди классов, но циклическое планирование внутри каждого класса. На рисунке 4 представлена система с четырьмя классами приоритетов. Алгоритм планирования выглядит следующим образом: пока в классе 4 есть готовые к запуску процессы, они запускаются один за другим согласно алгоритму циклического планирования, и каждому отводится квант времени. При этом классы с более низким приоритетом не будут их беспокоить. Если в классе 4 нет готовых к запуску процессов, запускаются процессы класса 3 и т. д. Если приоритеты постоянны, до процессов класса 1 процессор может не дойти никогда. Но за этим следит операционная система.

Аппаратный диспетчер задач при приоритетном планировании с классами таким же образом выбирает процесс с наивысшим значением в поле

PRIORITY, как и при простом приоритетном планировании. Но, если таких процессов несколько выбор осуществляется уже не случайным образом, а по времени поступления задачи. На то какой по счету задача поступила в очередь, указывает поле COUNTER. При этом за значение в поле PRIORITY продолжает отвечать операционная система, в то время как аппаратный диспетчер задач фиксирует порядок поступления задачи в поле COUNTER. Сначала выбирается задача с наивысшим приоритетом, а затем если таких несколько выбирается та задача, которая стоит в начале списка, другими словами, поступила раньше остальных задач с тем же приоритетом.



Рисунок 4 – Приоритетный алгоритм планирования с четырьмя классами приоритетов.

Вывод. Следуя вышесказанному, сделаем вывод, что аппаратная реализация алгоритмов планирования позволяет значительно быстрее определять, какая задача из списка готовых должна получить процессорное время следующей, нежели программная реализация тех же самых алгоритмов внутри планировщика операционной системы. Но какая бы дисциплина планирования не была выбрана, аппаратный диспетчер задач всегда работает в тесном взаимодействии с программным планировщиком, так как реализует лишь некоторые функции планирования и диспетчеризации, связанные с переключением процессов. Все остальные действия выполняет программный планировщик операционной системы.

Работа выполнена при финансовой поддержке стипендии Президента РФ молодым ученым и аспирантам на 2018-2020 гг. (СП-68.2018.5).

ЛИТЕРАТУРА

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 599 с.
2. Вашкевич Н.П. Аппаратная реализация функций синхронизации параллельных процессов при обращении к разделяемому ресурсу на основе ПЛИС/ Н.П. Вашкевич,

Р.А. Бикташев, Е.И. Гурин // Известия вузов. Поволжский регион. Технические науки – 2007. – № 2. – С.3–12.

3. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э.Таненбаум, М. ван Стеен // СПб: Питер, 2003. – 877 с.

4. Мартышкин А.И. Исследование алгоритмов планирования процессов в системах реального времени // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIII Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2015. – С. 118-124.

Мартышкин А.И., Мартенс-Атюшев Д.С. Разработка подсистемы планирования и назначения задач реконфигурируемой вычислительной системы для цифровой обработки сигнала // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIV Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2016. – С. 115-119.

УДК 004.451

АППАРАТНАЯ ПОДДЕРЖКА И ПРОВЕРКА ПРАВИЛЬНОСТИ РАБОТЫ ДИСПЕТЧЕРА ЗАДАЧ МНОГОПРОЦЕССОРНОЙ РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

А.И. Мартышкин

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье представлена возможность аппаратной реализации и проверки правильности работы диспетчера задач многопроцессорной реконфигурируемой вычислительной системы. Приводится методика верификации в программе SMV. В ходе исследования получена кодировка недетерминированного автомата НДА на входном языке SMV в соответствии с описанной методикой.

Ключевые слова: реконфигурируемая вычислительная система, диспетчер задач, верификация алгоритмов, синхронизация процессов, операционная система, программа SMV.

HARDWARE SUPPORT AND VERIFICATION OF THE OPERATION OF THE TASKMAN CONTROLLER OF A MULTI-PROCESSOR RECONFIGURABLE COMPUTING SYSTEM

A.I. Martyshkin

Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The article presents the possibility of hardware implementation and verification of the correctness of the work of the task manager of a multiprocessor reconfigurable